Docket No.: K-0329 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	
Myung Sub SIM	:	
Serial No. 09/977,252	:	Group Art Unit: 2631
Confirm. No.: 2252	:	Examiner: Unassigned
Filed: October 16, 2001	:	
For: DEVICE AND METHOD FOR DECODING TURBO CODES	:	

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D. C. 20231

Sir:

Prior to initial examination on the merits, please amend the above-identified application as follows:

IN THE ABSTRACT:

Please replace the abstract with the attached abstract.

IN THE SPECIFICATION:

Please amend the specification by replacing paragraphs as follows:

A. Specification Paragraphs With Mark-ups to Show Changes Made

The following are mark-ups to show changes made to paragraph(s) starting at page 1, line 5 and ending at page 1, line 13:

As is well known, turbo codes are generated by two (or more) recursive systematic convolutional encoders (RSCs) connected in parallel through an internal interleaver, and this coding method is used for transmitting data of a high data rate in the next generation mobile communication standard (3GPP or 3GPP2).

The turbo code processes a generated information bit sequence in the unit of a block. Especially in case of encoding a large information bit sequence, it is known that a very superior coding gain is obtained with respect to the convolutional codes and a very superior error correction capability is achieved by iteratively decoding simple component codes [in] at a receiving end.

The following are mark-ups to show changes made to paragraph(s) starting at page 2, line 8 and ending at page 2, line 10:

Generally, an information source of the turbo codes is the "Markov process" that has a discontinuous time and quantified states. Accordingly, the information source can be expressed through a binary matrix diagram.

The following are mark-ups to show changes made to paragraph(s) starting at page 2, line 14 and ending at page 3, line 3:

When the time is shifted from k-1 to k, an input bit d_k of the turbo encoder changes the state [S_k-1] $\underline{S}_{k-1}$ of the encoder to S_k. A state sequence $S=(S_0,..,S_T)$ of the information starts at time k=0 and ends at time k=T. The initial state S_0 of the encoder is 0.

The output sequence x of the turbo encoder is modulated [to] by BPSK or QPSK and suffers fading in a discrete memory channel. Accordingly, the sequence received in the receiving end becomes $y=(y_1,k,y_k,k,y_T)$. Here, $y_k=(y_{k,1},k,y_{k,n})$.

As described above, the MAP algorithm is an algorithm for estimating the *a posteriori* probability of state shift of the information using the received sequence. The MAP algorithm calculates the *a posteriori* probability of information bits $P(d_k = 1 | y)$ and $P(d_k = 0 | y)$. Then, the output of the decoder can be finally obtained in the form of a [desired] log likelihood ratio (LLR), as expressed by equation 1.

The following are mark-ups to show changes made to paragraph(s) starting at page 3, line 9 and ending at page 3, line 14:

[Equation 2]

$$P(S_{k-1} = m', S_k = m, y) = P(S_{k-1} = m', y_{j<k}) P(y_{j>k} | S_k = m) P(S_k = [m'] \underline{m}, y_k | S_{k-1} = m')$$

In equation 2, $y_{j<k}$ represents the received sequence from the initial time to time k-1 and $y_{j>k}$ represents the received sequence from time k+1 to the last time.

Also in equation 2, $P(S_{k-1} = m', y_{j<k})$ is defined as $[\alpha(S_k\text{-}1)]$ $\underline{\alpha(S_{k-1})}$ and $[\alpha(S_k\text{-}1)]$ $\underline{\alpha(S_k)}$ is [defined] derived therefrom. $P(y_{j>k} \mid S_k = m)$ is defined as $\beta(S_k)$.

The following are mark-ups to show changes made to paragraph(s) starting at page 4, line 6 and ending at page 4, line 11:

FIG. 1 is a timing diagram of the related art MAP decoding. The X-axis represents the flow of time, and especially represents which symbol each processor processes as the time flows. The [number of symbols] symbol number of a forward processor is increased, and the [number of symbols] symbol number of a backward processor is [reduced] decreased. The slant-lined shading sections represent that the backward processor is in learning. The curved arrows represent the correlation between alpha 5,7 and beta 6,8 values required for the bit decision.

The following are mark-ups to show changes made to paragraph(s) starting at page 5, line 18 and ending at page 6, line 2:

Also, in case that only one backward processor is used for reducing [the amount of calculation] hardware resources, the decoding time is increased twice as much.

Also, if the learning process, forward processing, and backward processing are performed for the length of L, the characteristic that the coding is completed with [ST=0] $\underline{S_T = 0}$ at a trellis termination of the turbo codes cannot be sufficiently used. This causes the coding gain of the turbo codes to deteriorate.

The following are mark-ups to show changes made to paragraph(s) starting at page 6, line 16 and ending at page 8, line 2:

A further object of the present invention is to provide a device and method of decoding turbo codes that is suitable for increasing a coding gain.

A further object of the present invention is to provide a device and method for decoding turbo codes that reduces the power consumption.

A further object of the present invention is to provide a device and method for decoding turbo codes trellis termination at maximum[, using one backward processor].

A further object of the present invention is to provide a device and method of decoding turbo codes that is suitable for reducing a decoding time.

To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, [a] an MAP decoder [in] at a receiving end that performs an iterative decoding includes a backward processor for calculating first resultant values that are state probability values, after a reference number of a received sequence for an L bit length of the received sequence, and calculating and storing second resultant values that are state probability values, after the reference number of the received sequence for a W bit length of a next received sequence; a forward processor for calculating third resultant values that are state probability values, before the reference number of the received sequence, simultaneously with calculation of the first resultant values of the next window; a memory for storing in order the second resultant values and outputting in a reverse order the second

resultant values [after calculation of the second resultant values]; and an output determination module for determining output values of the received sequence using the calculated third resultant values and the outputted second resultant values of the previous window.

Preferably, the memory [writes] stores the second resultant values by alternately using increasing addresses and decreasing addresses and outputs the second resultant values by alternately using the decreasing addresses in symmetry with the write and the increasing addresses.

The following are mark-ups to show changes made to paragraph(s) starting at page 8, line 14 and ending at page 9, line 3:

In another aspect of the present invention, a method of performing [a] an MAP turbo decoding [in] at a receiving end that performs an iterative decoding, includes the steps of calculating first resultant values that are state probability values after a reference number of a received sequence for an L bit length of the received sequence and calculating and storing second resultant values that are state probability values after the reference number of the received sequence for a W bit length of a next received sequence; calculating third resultant values that are state probability values before the reference number of the received sequence simultaneously with calculation of the first resultant values of the next window; storing in order the second resultant values and outputting in a reverse order the second resultant values [after calculation of the second resultant values]; and determining output values of the received

sequence using the calculated third resultant values and the outputted second resultant values of the previous window.

The following are mark-ups to show changes made to paragraph(s) starting at page 9, line 15 and ending at page 9, line 17:

Accordingly, the present invention considers the characteristic of trellis termination at the [termination] last part of the MAP [decoder] decoding and, thus, has an effect of [obtaining] increasing the coding gain.

The following are mark-ups to show changes made to paragraph(s) starting at page 10, line 1 and ending at page 11, line 4:

In still another aspect of the present invention, a method of decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, includes the steps of performing a learning by a backward processing for a predetermined length, calculating and storing [first] second resultant values by the backward processing, calculating [second] third resultant values by a forward processing simultaneously with the learning time of the next window, and determining a [decoding] decoded bit output using the [second] third resultant values and the [first] second resultant values of the previous window stored before the [first] second resultant values.

Preferably, if a length of the backward or forward processing is W, a length of learning is L, a remainder obtained by dividing the length of the received sequence by W is W_0, and N

is an integer not less than 1, the learning is performed by the backward processing with symbols of the [number of] received sequence number W_0+NW+L to W_0+NW, the [first] second resultant values by the backward processing with the symbols of W_0+NW to W_0+(N-1)W are stored, the [second] third resultant values by the forward processing with the symbols from W_0+(N-1)W[+L] to W_0+NW are calculated simultaneously with the next learning time, and a decoding bit determination is performed with the [second] third resultant values and the [first] second resultant values calculated and stored from W_0+(N-1)W to W_0+NW.

However, in case that N is 0, the learning is performed by the backward processing with the symbols of the number of received sequence W_0+L to W_0, the [first] second resultant values by the backward processing with the symbols of W_0 to 0 are stored and then the [second] third resultant values by the forward processing with the symbols from 0 to W_0 are calculated simultaneously with the learning start of [a] the next window to calculate the second resultant values of the next window. Here, the [first] second resultant values are written through one port of a dual-port RAM (DPRAM) and read out through another port thereof. Addresses stored or read out through the ports of the DPRAM are increased or decreased for each length of W_0 or W and the addresses stored or read out through the ports of the DPRAM are [increased] decreased or [decreased] increased for each length of W_0 or W in a mutually exclusive manner.

The following are mark-ups to show changes made to paragraph(s) starting at page 11, line 6 and ending at page 12, line 13:

The objects of the present invention may be achieved in whole or in part by [a] an MAP decoder, including a backward processor that calculates first resultant values of an L-bit [length] long sequence and second resultant values of a W-bit [length] long sequence; a forward processor that calculates third resultant values; a memory that stores the second resultant values in a first order and outputs the second resultant values in a second order that is the reverse of the first order; and an output determination module that determines output values of a received sequence using the third resultant values and the outputted second resultant values of the previous window. The L-bit [length] long sequence and W-bit [length] long sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

The objects of the present invention may be further achieved in whole or in part by a method of performing [a] an MAP turbo decoding. The method includes calculating first resultant values of an L-bit [length] long sequence, calculating second resultant values of a W-bit [length] long sequence, calculating third resultant values, storing the second resultant values in a first order and outputting the stored second resultant values in a second order that is the reverse of the first order, and outputting decoded values of a received sequence using the third

resultant values and the outputted second resultant values of the previous window. The L-bit [length] long sequence and W-bit [length] long sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

The objects of the present invention may be further achieved in whole or in part by a method of turbo-decoding a received sequence using a Maximum A Posteriori (MAP) algorithm. The method includes performing a learning by a backward processing for a predetermined length, calculating and storing [first] second resultant values obtained by the backward processing, calculating [second] third resultant values by a forward processing that overlaps in time with the next learning, and determining a decoding bit output using the [second] third resultant values and the stored [first] second resultant values of the previous window.

The following are mark-ups to show changes made to paragraph(s) starting at page 13, line 16 and ending at page 14, line 5:

Specifically, referring to equation 2, the first term [($\alpha(S_k\text{-}1)$)] ($\alpha(S_{k\text{-}1})$) is a joint probability function of the state [S_k-1] $S_{k\text{-}1}$ that is m' in the received sequence having time indexes from 0 to k-1 and is expressed by the following equation 3.

[Equation 3]

$$[\alpha(S_k\text{-}1) = P(S_k\text{-}1 = m', y_{j<k})] \; \alpha(S_{k\text{-}1}) = P(S_{k\text{-}1} = m', y_{j<k})$$

In equation 3, $\alpha(S_k)$ is a joint probability [density] function of the state shift where the state S_{k-1} is m' and the state S_k is m in the received sequence $y_{j<k+1}$, having the sequence numbers from 0 to k, and is expressed by the following equation 4.

The following are mark-ups to show changes made to paragraph(s) starting at page 14, line 8 and ending at page 14, line 11:

The [second] third term in equation 2 $\gamma(S_{k-1},S_k)$ is a branch metric relating when a state S_{k-1} is shifted to a state S_k and is a conditional probability function where the next state S_k is m, on condition that the state S_{k-1} is m' and the sequence received at that time is y_k. It is expressed by the following equation 5.

The following are mark-ups to show changes made to paragraph(s) starting at page 14, line 14 and ending at page 14, line 16:

The [third] second term in equation 2 $\beta(S_k)$ is a conditional probability function where [the number of] the received sequence number $y_{j>k}$ is not less than k+1, on condition that the state S_k is m and is expressed by the following equation 6.

The following are mark-ups to show changes made to paragraph(s) starting at page 15, line 1 and ending at page 15, line 5:

In equation 6, d_k represents the information bit sequence before the turbo encoding and S_k represents the state (m={0,1,..,M-1}) of the encoder at the number k of the received sequence. [Both d_k and] S_k [have] has M kinds of states and the input bit d_k changes the state of the encoder from S_{k-1} to S_k when the number of the received sequences is shifted from k-1 to k.

The following are mark-ups to show changes made to paragraph(s) starting at page 16, line 7 and ending at page 16, line 11:

Simultaneously with the start of learning with the symbols from W_0+W+L to W_0+W, the forward processor calculates the alpha values 13 with the symbols from 0 to W_0 and determines the output bits of the decoder using the calculated alpha values 13 and the beta values 10 from W_0 to 0, which [are] were stored by the backward processor. The calculation of the alpha [13-15] 14-15 and beta [10-12] 11-12 values is performed for the length of W.

The following are mark-ups to show changes made to paragraph(s) starting at page 16, line 15 and ending at page 16, line 18:

Simultaneously with the start of learning with the symbols from W_0+2W+L to W_0+2W, the forward processor calculates the alpha values 14 with the symbols from W_0 to W_0+W and

determines the output bits of the decoder using the calculated alpha values 14 and the beta values 11 from W_0+W to W_0, which [are] were stored by the backward processor.

The following are mark-ups to show changes made to paragraph(s) starting at page 18, line 3 and ending at page 19, line 2:

If the addresses being used proceed always in the same direction (i.e., if it is repeated that the addresses of port A are decreased from W or W_0 to 0 and the addresses of port B are increased from 0 to W or W_0), the stored beta values are updated with new values before the stored beta values have been read out for use in determining the decoder output. In order to prevent this, the size of the memory for storing the beta values should be [increased twice] doubled or the method proposed according to the present invention should be used.

According to the present invention, one backward processing starts from W_0+NW+L (N=0,1,2,..) and this start point is increased by W for each subsequent sliding window. Also, the end point of the backward processing is W_0+NW (N=0,1,2,..) or 0, and is also increased by W. In performing the learning, symbols correspond to the L bits of the received sequence in the same manner as in the related art method.

W_0 is determined by the equation [N] T mod W, where [N] T is the length of the received sequence and "mod" [(]refers to modulo calculation[)]. However, if the result of the modulo calculation is 0, W is used instead.

For example, if the length of the received sequence is 3840 bits and W is 256, W_0 is determined to be 256, which is equal to W. If the length [N] T of the received sequence is 3841 bits, W_0 becomes 1. As W_0 is determined as described above, the final unit of the backward processing will be always W. By doing this, the property of the turbo codes that use the trellis termination (3GPP TS25.212 V2.2.1 Turbo Coding section, Oct.1999) such as 3GPP WCDMA can be effected at maximum.

The following are mark-ups to show changes made to paragraph(s) starting at page 20, line 1 and ending at page 20, line 2:

Also, since the decoding results can be obtained in order, the memory and the circuit required for the LIFO can be removed and the power consumption can be [improved] reduced.

B. Clean Specification Changes

Please replace paragraph(s) starting at page 1, line 5 and ending at page 1, line 13 with the following paragraph(s):

As is well known, turbo codes are generated by two (or more) recursive systematic convolutional encoders (RSCs) connected in parallel through an internal interleaver, and this coding method is used for transmitting data of a high data rate in the next generation mobile communication standard (3GPP or 3GPP2).

The turbo code processes a generated information bit sequence in the unit of a block. Especially in case of encoding a large information bit sequence, it is known that a very superior coding gain is obtained with respect to the convolutional codes and a very superior error correction capability is achieved by iteratively decoding simple component codes at a receiving end.

Please replace paragraph(s) starting at page 2, line 8 and ending at page 2, line 10 with the following paragraph(s):

Generally, an information source of the turbo codes is the "Markov process" that has a discontinuous time and quantified states. Accordingly, the information source can be expressed through a binary matrix diagram.

Please replace paragraph(s) starting at page 2, line 14 and ending at page 3, line 3 with the following paragraph(s):

When the time is shifted from k-1 to k, an input bit d_k of the turbo encoder changes the state S_{k-1} of the encoder to S_k. A state sequence $S=(S_0,..,S_T)$ of the information starts at time k=0 and ends at time k=T. The initial state S_0 of the encoder is 0.

The output sequence x of the turbo encoder is modulated by BPSK or QPSK and suffers fading in a discrete memory channel. Accordingly, the sequence received in the receiving end becomes $y=(y_1,k,y_k,k,y_T)$. Here, $y_k=(y_{k,1},k,y_{k,n})$.

As described above, the MAP algorithm is an algorithm for estimating the *a posteriori* probability of state shift of the information using the received sequence. The MAP algorithm calculates the *a posteriori* probability of information bits $P(d_k = 1 | y)$ and $P(d_k = 0 | y)$. Then, the output of the decoder can be finally obtained in the form of a log likelihood ratio (LLR), as expressed by equation 1.

Please replace paragraph(s) starting at page 3, line 9 and ending at page 3, line 14 with the following paragraph(s):

[Equation 2]

$$P(S_{k-1} = m', S_k = m, y) = P(S_{k-1} = m', y_{j<k}) \, P(y_{j>k} \mid S_k = m) \, P(S_k = m, y_k \mid S_{k-1} = m')$$

In equation 2, $y_{j<k}$ represents the received sequence from the initial time to time k-1 and $y_{j>k}$ represents the received sequence from time k+1 to the last time.

Also in equation 2, $P(S_{k-1} = m', y_{j<k})$ is defined as $\alpha(S_{k-1})$ and $\alpha(S_k)$ is derived therefrom. $P(y_{j>k} \mid S_k = m)$ is defined as $\beta(S_k)$.

Please replace paragraph(s) starting at page 4, line 6 and ending at page 4, line 11 with the following paragraph(s):

FIG. 1 is a timing diagram of the related art MAP decoding. The X-axis represents the flow of time, and especially represents which symbol each processor processes as the time flows. The symbol number of a forward processor is increased, and the symbol number of a backward processor is decreased. The slant-lined shading sections represent that the backward processor is in learning. The curved arrows represent the correlation between alpha 5,7 and beta 6,8 values required for the bit decision.

Please replace paragraph(s) starting at page 5, line 18 and ending at page 6, line 2 with the following paragraph(s):

Also, in case that only one backward processor is used for reducing hardware resources, the decoding time is increased twice as much.

Also, if the learning process, forward processing, and backward processing are performed for the length of L, the characteristic that the coding is completed with $S_T = 0$ at a trellis termination of the turbo codes cannot be sufficiently used. This causes the coding gain of the turbo codes to deteriorate.

Please replace paragraph(s) starting at page 6, line 16 and ending at page 8, line 2 with the following paragraph(s):

A further object of the present invention is to provide a device and method of decoding turbo codes that is suitable for increasing a coding gain.

A further object of the present invention is to provide a device and method for decoding turbo codes that reduces the power consumption.

A further object of the present invention is to provide a device and method for decoding turbo codes trellis termination at maximum.

A further object of the present invention is to provide a device and method of decoding turbo codes that is suitable for reducing a decoding time.

To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, an MAP decoder at a receiving end that performs an iterative decoding includes a backward processor for calculating first resultant values that are state probability values, after a reference number of a received sequence for an L bit length of the received sequence, and calculating and storing second resultant values that are state probability values, after the reference number of the received sequence for a W bit length of a next received sequence; a forward processor for calculating third resultant values that are state probability values, before the reference number of the received sequence, simultaneously with calculation of the first resultant values of the next window; a memory for storing in order the second resultant values and outputting in a reverse order the second

resultant values; and an output determination module for determining output values of the received sequence using the calculated third resultant values and the outputted second resultant values of the previous window.

Preferably, the memory stores the second resultant values by alternately using increasing addresses and decreasing addresses and outputs the second resultant values by alternately using the decreasing addresses in symmetry with the write and the increasing addresses.

Please replace paragraph(s) starting at page 8, line 14 and ending at page 9, line 3 with the following paragraph(s):

In another aspect of the present invention, a method of performing an MAP turbo decoding at a receiving end that performs an iterative decoding, includes the steps of calculating first resultant values that are state probability values after a reference number of a received sequence for an L bit length of the received sequence and calculating and storing second resultant values that are state probability values after the reference number of the received sequence for a W bit length of a next received sequence; calculating third resultant values that are state probability values before the reference number of the received sequence simultaneously with calculation of the first resultant values of the next window; storing in order the second resultant values and outputting in a reverse order the second resultant values; and determining output values of the received sequence using the calculated third resultant values and the outputted second resultant values of the previous window.

Please replace paragraph(s) starting at page 9, line 15 and ending at page 9, line 17 with the following paragraph(s):

Accordingly, the present invention considers the characteristic of trellis termination at the last part of the MAP decoding and, thus, has an effect of increasing the coding gain.

Please replace paragraph(s) starting at page 10, line 1 and ending at page 11, line 4 with the following paragraph(s):

In still another aspect of the present invention, a method of decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, includes the steps of performing a learning by a backward processing for a predetermined length, calculating and storing second resultant values by the backward processing, calculating third resultant values by a forward processing simultaneously with the learning time of the next window, and determining a decoded bit output using the third resultant values and the second resultant values of the previous window stored before the second resultant values.

Preferably, if a length of the backward or forward processing is W, a length of learning is L, a remainder obtained by dividing the length of the received sequence by W is W_0, and N is an integer not less than 1, the learning is performed by the backward processing with symbols of the received sequence number W_0+NW+L to W_0+NW, the second resultant values by the backward processing with the symbols of W_0+NW to $W_0+(N-1)W$ are stored, the third resultant values by the forward processing with the symbols from $W_0+(N-1)W$ to W_0+NW are calculated

simultaneously with the next learning time, and a decoding bit determination is performed with the third resultant values and the second resultant values calculated and stored from $W_0+(N-1)W$ to W_0+NW.

However, in case that N is 0, the learning is performed by the backward processing with the symbols of the number of received sequence W_0+L to W_0, the second resultant values by the backward processing with the symbols of W_0 to 0 are stored and then the third resultant values by the forward processing with the symbols from 0 to W_0 are calculated simultaneously with the learning start of the next window to calculate the second resultant values of the next window. Here, the second resultant values are written through one port of a dual-port RAM (DPRAM) and read out through another port thereof. Addresses stored or read out through the ports of the DPRAM are increased or decreased for each length of W_0 or W and the addresses stored or read out through the ports of the DPRAM are decreased or increased for each length of W_0 or W in a mutually exclusive manner.

Please replace paragraph(s) starting at page 11, line 6 and ending at page 12, line 13 with the following paragraph(s):

The objects of the present invention may be achieved in whole or in part by an MAP decoder, including a backward processor that calculates first resultant values of an L-bit long sequence and second resultant values of a W-bit long sequence; a forward processor that calculates third resultant values; a memory that stores the second resultant values in a first order

and outputs the second resultant values in a second order that is the reverse of the first order; and an output determination module that determines output values of a received sequence using the third resultant values and the outputted second resultant values of the previous window. The L-bit long sequence and W-bit long sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

The objects of the present invention may be further achieved in whole or in part by a method of performing an MAP turbo decoding. The method includes calculating first resultant values of an L-bit long sequence, calculating second resultant values of a W-bit long sequence, calculating third resultant values, storing the second resultant values in a first order and outputting the stored second resultant values in a second order that is the reverse of the first order, and outputting decoded values of a received sequence using the third resultant values and the outputted second resultant values of the previous window. The L-bit long sequence and W-bit long sequence are portions of the received sequence. The first, second, and third resultant values are state probability values. The calculations of the first and third resultant values overlap in time and the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

The objects of the present invention may be further achieved in whole or in part by a method of turbo-decoding a received sequence using a Maximum A Posteriori (MAP) algorithm.

The method includes performing a learning by a backward processing for a predetermined length, calculating and storing second resultant values obtained by the backward processing, calculating third resultant values by a forward processing that overlaps in time with the next learning, and determining a decoding bit output using the third resultant values and the stored second resultant values of the previous window.

Please replace paragraph(s) starting at page 13, line 16 and ending at page 14, line 5 with the following paragraph(s):

Specifically, referring to equation 2, the first term ($\alpha(S_{k-1})$) is a joint probability function of the state S_{k-1} that is m' in the received sequence having time indexes from 0 to k-1 and is expressed by the following equation 3.

[Equation 3]

$$\alpha(S_{k-1}) = P(S_{k-1} = m', y_{j<k})$$

In equation 3, $\alpha(S_k)$ is a joint probability function of the state shift where the state S_{k-1} is m' and the state S_k is m in the received sequence $y_{j<k+1}$, having the sequence numbers from 0 to k, and is expressed by the following equation 4.

Please replace paragraph(s) starting at page 14, line 8 and ending at page 14, line 11 with the following paragraph(s):

The third term in equation 2 $\gamma(S_{k-1},S_k)$ is a branch metric relating when a state S_{k-1} is shifted to a state S_k and is a conditional probability function where the next state S_k is m, on condition that the state S_{k-1} is m' and the sequence received at that time is y_k. It is expressed by the following equation 5.

Please replace paragraph(s) starting at page 14, line 14 and ending at page 14, line 16 with the following paragraph(s):

The second term in equation 2 $\beta(S_k)$ is a conditional probability function where the received sequence number $y_{j>k}$ is not less than k+1, on condition that the state S_k is m and is expressed by the following equation 6.

Please replace paragraph(s) starting at page 15, line 1 and ending at page 15, line 5 with the following paragraph(s):

In equation 6, d_k represents the information bit sequence before the turbo encoding and S_k represents the state (m={0,1,..,M-1}) of the encoder at the number k of the received sequence. S_k has M kinds of states and the input bit d_k changes the state of the encoder from S_{k-1} to S_k when the number of the received sequences is shifted from k-1 to k.

Please replace paragraph(s) starting at page 16, line 7 and ending at page 11, line 18 with the following paragraph(s):

Simultaneously with the start of learning with the symbols from W_0+W+L to W_0+W, the forward processor calculates the alpha values 13 with the symbols from 0 to W_0 and determines the output bits of the decoder using the calculated alpha values 13 and the beta values 10 from W_0 to 0, which were stored by the backward processor. The calculation of the alpha 14-15 and beta 11-12 values is performed for the length of W.

Please replace paragraph(s) starting at page 16, line 15 and ending at page 16, line 18 with the following paragraph(s):

Simultaneously with the start of learning with the symbols from W_0+2W+L to W_0+2W, the forward processor calculates the alpha values 14 with the symbols from W_0 to W_0+W and determines the output bits of the decoder using the calculated alpha values 14 and the beta values 11 from W_0+W to W_0, which were stored by the backward processor.

Please replace paragraph(s) starting at page 18, line 3 and ending at page 19, line 2 with the following paragraph(s):

If the addresses being used proceed always in the same direction (i.e., if it is repeated that the addresses of port A are decreased from W or W_0 to 0 and the addresses of port B are increased from 0 to W or W_0), the stored beta values are updated with new values before the

stored beta values have been read out for use in determining the decoder output. In order to prevent this, the size of the memory for storing the beta values should be doubled or the method proposed according to the present invention should be used.

According to the present invention, one backward processing starts from W_0+NW+L (N=0,1,2,..) and this start point is increased by W for each subsequent sliding window. Also, the end point of the backward processing is W_0+NW (N=0,1,2,..) or 0, and is also increased by W. In performing the learning, symbols correspond to the L bits of the received sequence in the same manner as in the related art method.

W_0 is determined by the equation T mod W, where T is the length of the received sequence and "mod" refers to modulo calculation. However, if the result of the modulo calculation is 0, W is used instead.

For example, if the length of the received sequence is 3840 bits and W is 256, W_0 is determined to be 256, which is equal to W. If the length T of the received sequence is 3841 bits, W_0 becomes 1. As W_0 is determined as described above, the final unit of the backward processing will be always W. By doing this, the property of the turbo codes that use the trellis termination (3GPP TS25.212 V2.2.1 Turbo Coding section, Oct.1999) such as 3GPP WCDMA can be effected at maximum.

Please replace paragraph(s) starting at page 20, line 1 and ending at page 20, line 2 with the following paragraph(s):

Also, since the decoding results can be obtained in order, the memory and the circuit required for the LIFO can be removed and the power consumption can be reduced.

IN THE CLAIMS:

A. Please amend claims 1, 2, 4, 5, 6, 7, 8, and 11-21 as follows:

1. (Amended) [A] An MAP decoder, comprising:

a backward processor that calculates first resultant values of an L-symbol [length] long sequence and second resultant values of a W-symbol [length] long sequence;

a forward processor that calculates third resultant values;

a memory that stores the second resultant values in a first order and outputs the second resultant values in a second order that is the reverse of the first order; and

an output determination module that determines output values of a received sequence using the third resultant values and the outputted second resultant values of the previous window, wherein

the L-symbol [length] long sequence and the W-symbol [length] long sequence are portions of the received sequence,

the first, second, and third resultant values are state probability values,

the calculations of the first and third resultant values overlap in time, and

the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

2. (Amended) The MAP decoder of claim 1, wherein the memory [writes] stores a plurality of groups of the second resultant values by alternately using increasing and decreasing sequential addresses to [write] store subsequent groups of the second resultant values.

4. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of [M] T symbols;

each of the first resultant values corresponds to an input value of the L-symbol [length] long sequence;

each of the first resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the first resultant values corresponds to the i^{th} symbol received in the received sequence, where i is identified by the equation [i = W + L + (M mod W) and (M mod W)] i = L + (T mod W) and (T mod W) is modulo division providing the remainder of the division.

5. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of [M] T symbols;

each of the second resultant values corresponds to an input value of the W-symbol [length] long sequence;

each of the second resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the second resultant values corresponds to the j^{th} symbol received in the received sequence, where j is identified by the equation [j = (M mod W) and (M mod W)] j = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

6. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of [M] T symbols;

each of the third resultant values corresponds to an input value of the W-symbol [length] long sequence;

each of the third resultant values is calculated in the order of receipt of the corresponding input value; and

a last calculated value in the ordered sequence of the third resultant values corresponds to the k^{th} symbol received in the received sequence, where k is identified by the equation [k = (M mod W) and (M mod W)] k = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

7. (Amended) The MAP decoder of claim 3, wherein:

each of the output values corresponds to an input value of the W-symbol [length] long sequence; and

the output determination module outputs each of the output values in the same order as the order of receipt of the corresponding input value of the W-symbol [length] long sequence.

8. (Amended) A method of performing [a] an MAP turbo decoding, comprising:

calculating first resultant values of an L-symbol [length] long sequence;

calculating second resultant values of a W-symbol [length] long sequence;

calculating third resultant values of the W-symbol [length] long sequence;

storing the second resultant values in a first order and outputting the stored second resultant values in a second order that is the reverse of the first order; and

outputting decoded values of a received sequence using the third resultant values and the outputted second resultant values of the previous window, wherein

the L-symbol [length] long sequence and W-symbol [length] long sequence are portions of the received sequence,

the first, second, and third resultant values are state probability values,

the calculations of the first and third resultant values overlap in time, and

the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

11. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of [M] T symbols;

each of the first resultant values corresponds to an input value of the L-symbol [length] long sequence;

each of the first resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the first resultant values corresponds to the i^{th} symbol received in the received sequence, where i is identified by the equation [i = W + L + (M mod W) and (M mod W)] i = L + (T mod W) and (T mod W) is modulo division providing the remainder of the division.

12. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of [M] T symbols;

each of the second resultant values corresponds to an input value of the W-symbol [length] long sequence;

each of the second resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the second resultant values corresponds to the j^{th} symbol received in the received sequence, where j is identified by the equation [j = (M

mod W) and (M mod W)] j = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

13. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of [M] T symbols;

the W-symbol [length] long sequence is a portion of the received sequence;

each of the third resultant values corresponds to an input value of the W-symbol [length] long sequence;

each of the third resultant values is calculated in the order of receipt of the corresponding input value; and

a last calculated value in the ordered sequence of the third resultant values corresponds to the k^{th} symbol received in the received sequence, where k is identified by the equation [k = (M mod W) and (M mod W)] k = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

14. (Amended) The method of claim 8, wherein:

each of the decoded values corresponds to an input value of the W-symbol [length] long sequence; and

the decoded values are output in the same order as the order of receipt of the corresponding input value of the W-symbol [length] long sequence.

15. (Amended) A method of turbo-decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, comprising:

performing a learning by a backward processor for a predetermined length;

calculating and storing [first] second resultant values obtained by the backward processor;

calculating [second] third resultant values by a forward processor that overlaps in time with the next learning; and

determining a decoding symbol output using the [second] third resultant values and the stored [first] second resultant values of the previous window.

16. (Amended) The turbo-decoding method of claim 15, wherein if a processing length of the backward or forward processor is W, a length of learning is L, a remainder obtained by dividing a length of a received sequence by W is W_0, and N is an integer not less than 1,

the learning is performed by the backward processor on sequential symbol portions of the received sequence identified by the range W_0+NW+L to W_0+NW;

the [first] second resultant values calculated by the backward processor on sequential symbol portions identified by the range of the received sequence W_0+NW to $W_0+(N-1)W$ are stored;

the [second] third resultant values calculated by the forward processor on the sequential symbol portions of the received sequence identified by $W_0+(N-1)W$ to W_0+NW; and

the decoding symbol determination is performed with the [second] third resultant values and the [first] second resultant values based on the sequential symbol portions of the received sequence identified by the range $W_0+(N-1)W$ to W_0+NW.

17. (Amended) The turbo-decoding method of claim 15, wherein if a processing length of the backward or forward processor is W, a length of learning is L, a remainder obtained by dividing a length of the received sequence by W is W_0, and N is an integer equal to 0;

the learning is performed by the backward processor on sequential symbol portions of the received sequence identified by the range [W_0+NW+L to W_0+NW] W_0+L to W_0;

the [first] second resultant values calculated by the backward processor on sequential symbol portions of the received sequence identified by the range W_0[+NW] to 0 are stored; and

the [second] third resultant values calculated by the forward processor on the sequential symbol portions of the received sequence identified by the range 0 to W_0[+NW] are calculated during a period overlapping in time simultaneously the learning performed in [a] the next window.

18. (Amended) The turbo-decoding method of claim 15, wherein the [first] second resultant values are written through one port of a dual-port RAM (DPRAM) and are read out through another port thereof.

19. (Amended) The method of claim 15, further comprising writing a plurality of groups of the [first] second resultant values by alternately using increasing and decreasing sequential addresses to store subsequent groups of the [first] second resultant values.

20. (Amended) The method of claim 15, further comprising:

repeatedly storing newly calculated [first] second resultant values and reading the stored second resultant values by alternately using: (1) increasing sequential addresses for the store operation and decreasing sequential addresses for the read operation and (2) decreasing sequential addresses for the store operation and increasing sequential addresses for the read operation, as the write and read operations are applied to each of a plurality of groups of the [first] second resultant values.

21. (Amended) The method of claim 15, further comprising:

outputting a plurality of the decoding symbol outputs as decoded values, wherein

each of the decoded values corresponds to an input value of a W-symbol [length] long sequence; and

the decoded values are output in the same order as the order of receipt of the corresponding input value of the W-symbol [length] long sequence.

Clean Set of Amended Claims

1. (Amended) An MAP decoder, comprising:

a backward processor that calculates first resultant values of an L-symbol long sequence and second resultant values of a W-symbol long sequence;

a forward processor that calculates third resultant values;

a memory that stores the second resultant values in a first order and outputs the second resultant values in a second order that is the reverse of the first order; and

an output determination module that determines output values of a received sequence using the third resultant values and the outputted second resultant values of the previous window, wherein

the L-symbol long sequence and the W-symbol long sequence are portions of the received sequence,

the first, second, and third resultant values are state probability values,

the calculations of the first and third resultant values overlap in time, and

the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

2. (Amended) The MAP decoder of claim 1, wherein the memory stores a plurality of groups of the second resultant values by alternately using increasing and decreasing sequential addresses to store subsequent groups of the second resultant values.

4. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of T symbols;

each of the first resultant values corresponds to an input value of the L-symbol long sequence;

each of the first resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the first resultant values corresponds to the i^{th} symbol received in the received sequence, where i is identified by the equation i = L + (T mod W) and (T mod W) is modulo division providing the remainder of the division.

5. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of T symbols;

each of the second resultant values corresponds to an input value of the W-symbol long sequence;

each of the second resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the second resultant values corresponds to the j^{th} symbol received in the received sequence, where j is identified by the equation j = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

6. (Amended) The MAP decoder of claim 1, wherein:

the received sequence has a symbol length of T symbols;

each of the third resultant values corresponds to an input value of the W-symbol long sequence;

each of the third resultant values is calculated in the order of receipt of the corresponding input value; and

a last calculated value in the ordered sequence of the third resultant values corresponds to the k^{th} symbol received in the received sequence, where k is identified by the equation k = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

7. (Amended) The MAP decoder of claim 3, wherein:

each of the output values corresponds to an input value of the W-symbol long sequence; and

the output determination module outputs each of the output values in the same order as the order of receipt of the corresponding input value of the W-symbol long sequence.

8. (Amended) A method of performing an MAP turbo decoding, comprising:

calculating first resultant values of an L-symbol long sequence;

calculating second resultant values of a W-symbol long sequence;

calculating third resultant values of the W-symbol long sequence;

storing the second resultant values in a first order and outputting the stored second resultant values in a second order that is the reverse of the first order; and

outputting decoded values of a received sequence using the third resultant values and the outputted second resultant values of the previous window, wherein

the L-symbol long sequence and W-symbol long sequence are portions of the received sequence,

the first, second, and third resultant values are state probability values,

the calculations of the first and third resultant values overlap in time, and

the calculation of the first resultant values for the next window is performed after the calculation of the second resultant values is completed.

11. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of T symbols;

each of the first resultant values corresponds to an input value of the L-symbol long sequence;

each of the first resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the first resultant values corresponds to the i^{th} symbol received in the received sequence, where i is identified by the equation $i = L + (T \bmod W)$ and $(T \bmod W)$ is modulo division providing the remainder of the division.

12. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of T symbols;

each of the second resultant values corresponds to an input value of the W-symbol long sequence;

each of the second resultant values is calculated in a sequential order that is the reverse of the order of receipt of the corresponding input value; and

a first calculated value in the ordered sequence of the second resultant values corresponds to the j^{th} symbol received in the received sequence, where j is identified by the equation j = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

13. (Amended) The method of claim 8, wherein:

the received sequence has a symbol length of T symbols;

the W-symbol long sequence is a portion of the received sequence;

each of the third resultant values corresponds to an input value of the W-symbol long sequence;

each of the third resultant values is calculated in the order of receipt of the corresponding input value; and

a last calculated value in the ordered sequence of the third resultant values corresponds to the k^{th} symbol received in the received sequence, where k is identified by the equation k = (T mod W) and (T mod W) is modulo division providing the remainder of the division.

14. (Amended) The method of claim 8, wherein:

each of the decoded values corresponds to an input value of the W-symbol long sequence; and

the decoded values are output in the same order as the order of receipt of the corresponding input value of the W-symbol long sequence.

15. (Amended) A method of turbo-decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, comprising:

performing a learning by a backward processor for a predetermined length;

calculating and storing second resultant values obtained by the backward processor;

calculating third resultant values by a forward processor that overlaps in time with the next learning; and

determining a decoding symbol output using the third resultant values and the stored second resultant values of the previous window.

16. (Amended) The turbo-decoding method of claim 15, wherein if a processing length of the backward or forward processor is W, a length of learning is L, a remainder obtained by dividing a length of a received sequence by W is W_0, and N is an integer not less than 1,

the learning is performed by the backward processor on sequential symbol portions of the received sequence identified by the range W_0+NW+L to W_0+NW;

the second resultant values calculated by the backward processor on sequential symbol portions identified by the range of the received sequence W_0+NW to W_0+(N-1)W are stored;

the third resultant values calculated by the forward processor on the sequential symbol portions of the received sequence identified by W_0+(N-1)W to W_0+NW; and

the decoding symbol determination is performed with the third resultant values and the second resultant values based on the sequential symbol portions of the received sequence identified by the range W_0+(N-1)W to W_0+NW.

17. (Amended) The turbo-decoding method of claim 15, wherein if a processing length of the backward or forward processor is W, a length of learning is L, a remainder obtained by dividing a length of the received sequence by W is W_0, and N is an integer equal to 0;

the learning is performed by the backward processor on sequential symbol portions of the received sequence identified by the range W_0 + L to W_0;

the second resultant values calculated by the backward processor on sequential symbol portions of the received sequence identified by the range W_0 to 0 are stored; and

the third resultant values calculated by the forward processor on the sequential symbol portions of the received sequence identified by the range 0 to W_0 are calculated during a period overlapping in time simultaneously the learning performed in the next window.

18. (Amended) The turbo-decoding method of claim 15, wherein the second resultant values are written through one port of a dual-port RAM (DPRAM) and are read out through another port thereof.

19. (Amended) The method of claim 15, further comprising writing a plurality of groups of the second resultant values by alternately using increasing and decreasing sequential addresses to store subsequent groups of the second resultant values.

20. (Amended) The method of claim 15, further comprising:

repeatedly storing newly calculated second resultant values and reading the stored second resultant values by alternately using: (1) increasing sequential addresses for the store operation and decreasing sequential addresses for the read operation and (2) decreasing sequential addresses for the store operation and increasing sequential addresses for the read operation, as the write and read operations are applied to each of a plurality of groups of the second resultant values.

21. (Amended) The method of claim 15, further comprising:

outputting a plurality of the decoding symbol outputs as decoded values, wherein each of the decoded values corresponds to an input value of a W-symbol long sequence; and

the decoded values are output in the same order as the order of receipt of the corresponding input value of the W-symbol long sequence.

REMARKS

Claims 1-21 are pending. Prompt examination and allowance in due course are respectfully solicited.

An amended Abstract of the Disclosure is enclosed herewith as a separate attachment.

Respectfully submitted,
FLESHNER & KIM, LLP

Daniel Y.J. Kim
Registration No. 36,186
David W. Ward
Registration No. 45,198

P.O. Box 221200
Chantilly, VA 20153-1200
703 502-9440 DYK/DWW:tmd
Date: January 14, 2002

ABSTRACT OF THE DISCLOSURE

A method and apparatus of decoding turbo codes using a sliding window method is disclosed. In decoding a received sequence using a Maximum A Posteriori (MAP) algorithm, a learning by a backward processing is performed for a predetermined length and second resultant values calculated by the backward processing are stored. Third resultant values calculated by a forward processing. The calculation of the third resultant values overlaps in time with the next learning. A decoding symbol output is determined using the third resultant values and the stored first resultant values of the previous window.